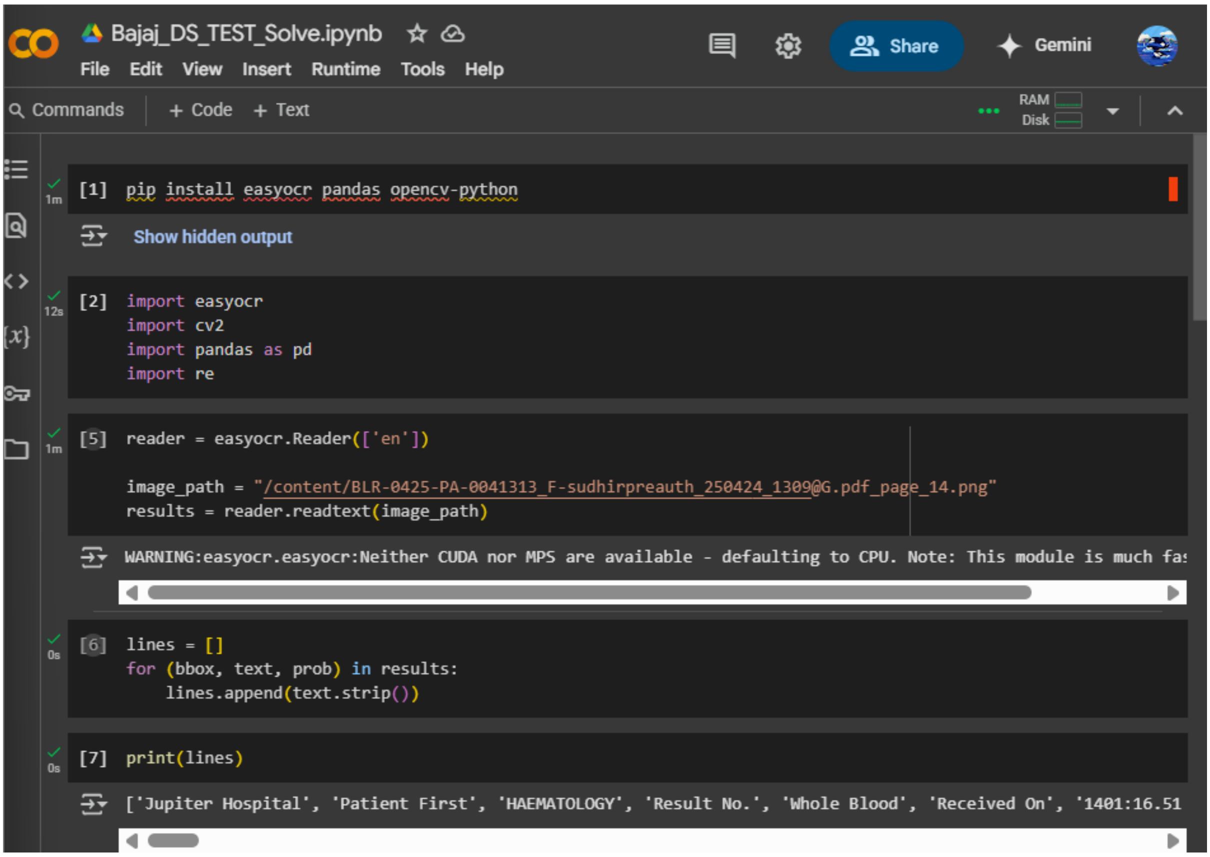
Name: Chanchal Vishwakarma [RA2211028010027]

Data Science

Git: https://github.com/Chanchal2411/Bajaj_Finserv_Health_DataScience_Qualifer2_solution


Bajaj_DS_TEST_Solve.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
RAM
Disk
Show hidden output

```
[1] pip install easyocr pandas opencv-python
```

```
[2] import easyocr
    import cv2
    import pandas as pd
    import re
```

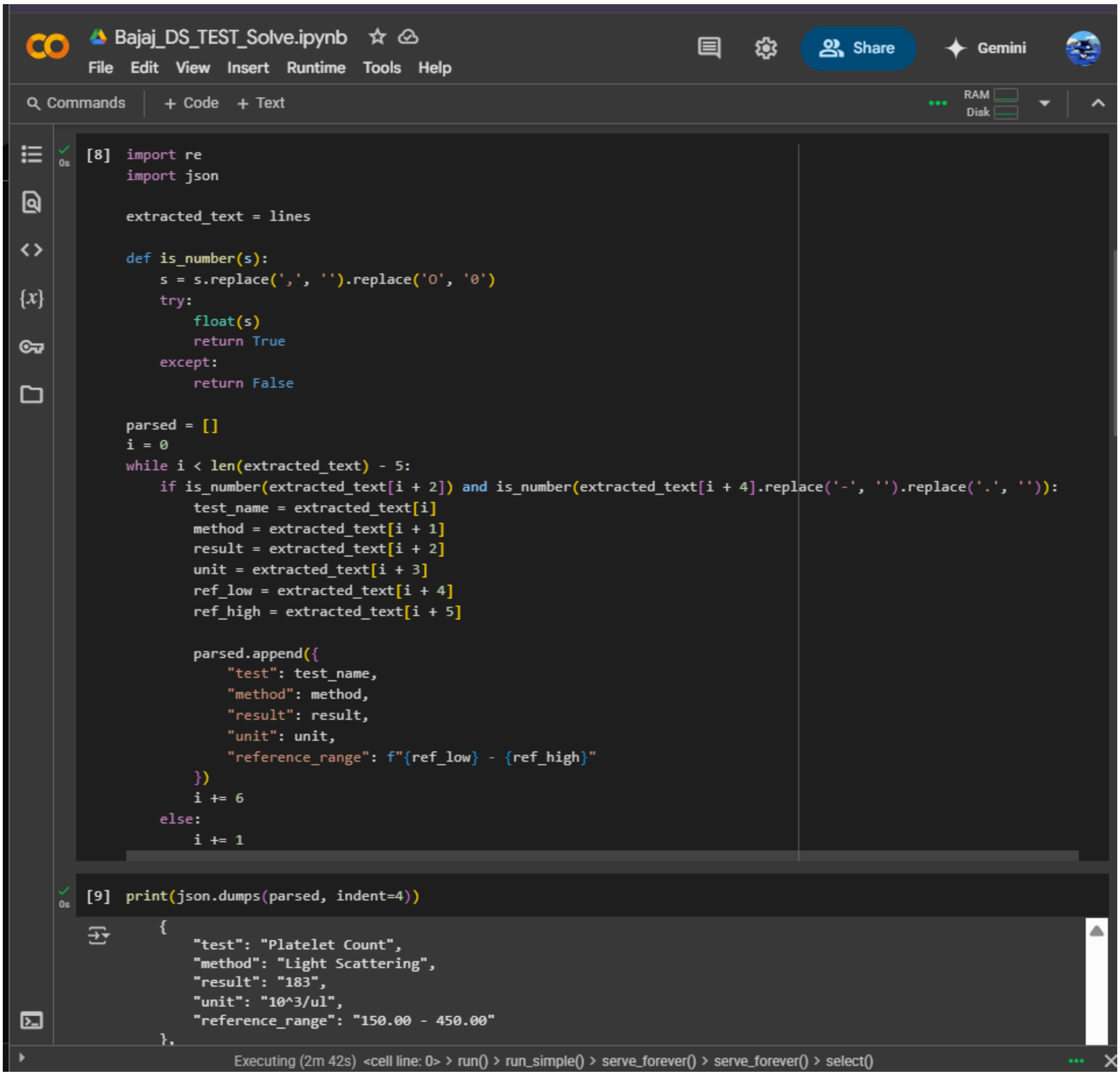
```
[5] reader = easyocr.Reader(['en'])

    image_path = "/content/BLR-0425-PA-0041313_F-sudhirpreauth_250424_1309@G.pdf_page_14.png"
    results = reader.readtext(image_path)
```

```
WARNING:easyocr.easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much fa:
```

```
[6] lines = []
    for (bbox, text, prob) in results:
        lines.append(text.strip())
```

```
[7] print(lines)
```

```
['Jupiter Hospital', 'Patient First', 'HAEMATOLOGY', 'Result No.', 'Whole Blood', 'Received On', '1401:16.51
```



[8]

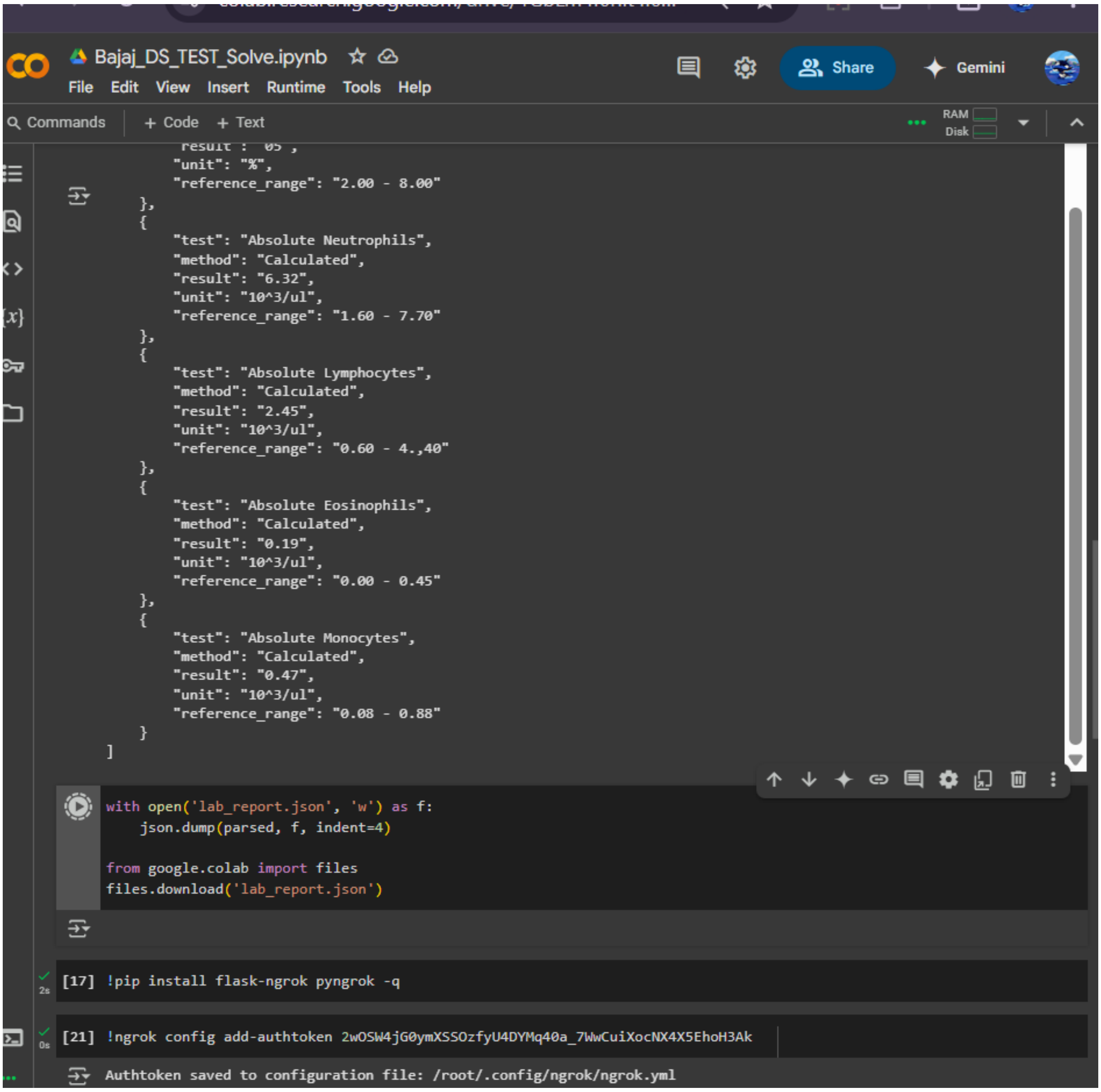
```
import re
import json

extracted_text = lines

def is_number(s):
    s = s.replace(',', '').replace('O', '0')
    try:
        float(s)
        return True
    except:
        return False

parsed = []
i = 0
while i < len(extracted_text) - 5:
    if is_number(extracted_text[i + 2]) and is_number(extracted_text[i + 4].replace('-', '').replace('.', '')):
        test_name = extracted_text[i]
        method = extracted_text[i + 1]
        result = extracted_text[i + 2]
        unit = extracted_text[i + 3]
        ref_low = extracted_text[i + 4]
        ref_high = extracted_text[i + 5]

        parsed.append({
            "test": test_name,
            "method": method,
            "result": result,
            "unit": unit,
            "reference_range": f"{ref_low} - {ref_high}"
        })
        i += 6
    else:
        i += 1
```

[9]

```
print(json.dumps(parsed, indent=4))
```

```
    {
        "test": "Platelet Count",
        "method": "Light Scattering",
        "result": "183",
        "unit": "10^3/ul",
        "reference_range": "150.00 - 450.00"
    },
```

Executing (2m 42s) <cell line: 0> > run() > run_simple() > serve_forever() > serve_forever() > select()



```
        "result": "05",
        "unit": "%",
        "reference_range": "2.00 - 8.00"
    },
    {
        "test": "Absolute Neutrophils",
        "method": "Calculated",
        "result": "6.32",
        "unit": "10^3/ul",
        "reference_range": "1.60 - 7.70"
    },
    {
        "test": "Absolute Lymphocytes",
        "method": "Calculated",
        "result": "2.45",
        "unit": "10^3/ul",
        "reference_range": "0.60 - 4.,40"
    },
    {
        "test": "Absolute Eosinophils",
        "method": "Calculated",
        "result": "0.19",
        "unit": "10^3/ul",
        "reference_range": "0.00 - 0.45"
    },
    {
        "test": "Absolute Monocytes",
        "method": "Calculated",
        "result": "0.47",
        "unit": "10^3/ul",
        "reference_range": "0.08 - 0.88"
    }
]
```

```python
with open('lab_report.json', 'w') as f:
    json.dump(parsed, f, indent=4)

from google.colab import files
files.download('lab_report.json')
```

```
2s [17] !pip install flask-ngrok pyngrok -q
```

```
0s [21] !ngrok config add-authtoken 2wOSW4jG0ymXSSOzfyU4DYMq40a_7WwCuiXocNX4X5EhoH3Ak
```

```
Authtoken saved to configuration file: /root/.config/ngrok/ngrok.yml
```